

[CLAIMS]

What is claimed is:

1. A method comprising:

5 reporting a spanning tree to a distributed dictionary for a plurality of nodes comprising a network, said plurality of nodes to adopt the spanning tree from the distributed dictionary in a coordinated manner to avoid a transient topology loop in the network.

10 2. The method of claim 1 wherein reporting the spanning tree comprises:

15 16 performing at least one iteration of a reporting task to disable links to be removed from among the plurality of nodes; and

20 17 performing at least one additional iteration of the reporting task to enable links to be added among the plurality of nodes only after the links to be removed have been disabled.

25 18 3. The method of claim 2 wherein the at least one iteration of the reporting task comprises:

20 identifying the links to be removed based on a comparison of the spanning tree to a previous spanning tree;

25 removing the links to be removed from the previous spanning tree to generate a modified spanning tree; and

30 registering the modified spanning tree to the distributed dictionary.

4. The method of claim 3 wherein registering the modified spanning tree comprises:

18 identifying an incarnation identifier for the previous spanning tree;

advancing the incarnation identifier; and

5 combining the incarnation identifier with the modified spanning tree.

5. The method of claim 3 wherein registering the modified spanning tree comprises:

15 storing the modified spanning tree to local memory; and

multicasting the spanning tree to a remainder of the plurality of nodes.

6. The method of claim 2 wherein the at least one additional iteration of the reporting

10 task comprises:

identifying the links to be added based on a comparison of the spanning tree to a previous spanning tree;

15 verifying there are no remaining links to be removed;

verifying that the plurality of nodes are synchronized with respect to adoption of the previous spanning tree;

20 adding the links to be added to the previous spanning tree to generate a modified spanning tree; and

registering the modified spanning tree to the distributed dictionary.

7. The method of claim 6 wherein verifying there are no remaining links comprises:

21

subtracting a set of links defined by the spanning tree from a set of links defined

by the previous spanning tree to provide a result;

verifying that the result is a null set;

accessing a set of acknowledgements from the plurality of nodes registered in

5 the distributed dictionary; and

verifying that an incarnation identifier for each acknowledgement in the set of
acknowledgements is equal.

8. The method of claim 2 wherein prior to performing a next iteration of the reporting

10 task, the method further comprises:

accessing acknowledgements from the plurality of nodes in the distributed
dictionary, said acknowledgements including incarnation identifiers for a most current
spanning tree adopted by respective ones of the plurality of nodes;

identifying a lowest incarnation identifier stored in the distributed dictionary;

15 comparing the lowest incarnation identifier to an incarnation identifier of a most
recently reported spanning tree; and

initiating the next iteration of the reporting task if the lowest incarnation identifier
is equal to the incarnation identifier of the most recently reported spanning tree.

8. A method comprising:

23 adopting a spanning tree from a distributed dictionary in a manner coordinated throughout a plurality of nodes comprising a network to avoid a transient topology loop in the network.

5

10. The method of claim 9 wherein adopting the spanning tree comprises:

24 identifying the spanning tree as a new spanning tree in the distributed dictionary; extracting ports from the new spanning tree corresponding to a particular node; disabling any ports at the particular node not extracted from the new spanning tree; and enabling any ports at the particular node after disabling any ports not extracted from the new spanning tree.

15 11. The method of claim 9 further comprising:

25 acknowledging an adoption of the spanning tree.

12. The method of claim 11 wherein acknowledging adoption of the spanning tree

26 comprises:

registering an incarnation identifier of a most currently adopted spanning tree to 20 the distributed dictionary.

13. The method of claim 10 wherein identifying the spanning tree as the new spanning tree comprises:

receiving the spanning tree at the particular node, said spanning tree including an incarnation identifier;

retrieving an incarnation identifier for an entry in the distributed dictionary corresponding to the spanning tree;

5 comparing the incarnation identifier of the spanning tree to the incarnation identifier for the entry; and

identifying the spanning tree as a new spanning tree if the incarnation identifier for the entry is older than the incarnation identifier for the spanning tree.

10 14. The method of claim 9 wherein adopting the spanning tree comprises:

2 14. performing at least one iteration of an adoption task to disable any ports corresponding to links to be removed from a previous spanning tree; and

15 14. performing at least one iteration of the adoption task to enable any ports corresponding to links to be added from the previous spanning tree only after the links to be removed have been disabled.

15 An article comprising:

a machine readable storage medium having stored thereon executable instructions to implement reporting a spanning tree to a distributed dictionary for a plurality of nodes comprising a network, said plurality of nodes to adopt the spanning tree from the distributed dictionary in a coordinated manner to avoid a transient topology loop in the network.

16. The article of claim 15 wherein the reporting the spanning tree comprises:

10 performing at least one iteration of a reporting task to disable links to be removed from among the plurality of nodes; and
15 performing at least one additional iteration of the reporting task to enable links to be added among the plurality of nodes only after the links to be removed have been disabled.

15 17. The article of claim 16 wherein the at least one iteration of the reporting task comprises:

identifying the links to be removed based on a comparison of the spanning tree to a previous spanning tree;
removing the links to be removed from the previous spanning tree to generate a
20 modified spanning tree; and
registering the modified spanning tree to the distributed dictionary.

18. The article of claim 17 wherein registering the modified spanning tree comprises:

identifying an incarnation identifier for the previous spanning tree;
advancing the incarnation identifier; and
combining the incarnation identifier with the modified spanning tree.

5 19. The article of claim 17 wherein registering the modified spanning tree comprises:

storing the modified spanning tree to local memory; and

multicasting the spanning tree to a remainder of the plurality of nodes.

10 20. The article of claim 16 wherein the at least one additional iteration of the reporting task comprises:

identifying the links to be added based on a comparison of the spanning tree to a previous spanning tree;

verifying there are no remaining links to be removed;

verifying that the plurality of nodes are synchronized with respect to adoption of

15 the previous spanning tree;

adding the links to be added to the previous spanning tree to generate a modified spanning tree; and

registering the modified spanning tree to the distributed dictionary.

20 21. The article of claim 20 wherein verifying there are no remaining links comprises:

subtracting a set of links defined by the spanning tree from a set of links defined by the previous spanning tree to provide a result;

verifying that the result is a null set;

accessing a set of acknowledgements from the plurality of nodes registered in the distributed dictionary; and verifying that an incarnation identifier for each acknowledgement in the set of acknowledgements is equal.

5 22. The article of claim 15 wherein prior to performing a next iteration of the reporting task, the executable instructions further implement:

10 accessing acknowledgements from the plurality of nodes in the distributed dictionary, said acknowledgements including incarnation identifiers for a most current spanning tree adopted by respective ones of the plurality of nodes;

15 identifying a lowest incarnation identifier stored in the distributed dictionary; comparing the lowest incarnation identifier to an incarnation identifier of a most recently reported spanning tree; and

initiating the next iteration of the reporting task if the lowest incarnation identifier

15 is equal to the incarnation identifier of the most recently reported spanning tree.

23. An article comprising:

a machine readable storage medium having stored thereon executable instructions to implement adopting a spanning tree from a distributed dictionary in a manner coordinated throughout a plurality of nodes comprising a network to avoid a 5 transient topology loop in the network.

24. The article of claim 23 wherein adopting the spanning tree comprises:

identifying the spanning tree as a new spanning tree in the distributed dictionary; extracting ports from the new spanning tree corresponding to a particular node; disabling any ports at the particular node not extracted from the new spanning tree; and enabling any ports at the particular node after disabling any ports not extracted from the new spanning tree.

15— 25. The article of claim 23 wherein the executable instructions further implement:
acknowledging an adoption of the spanning tree.

26. The article of claim 25 wherein acknowledging adoption of the spanning tree comprises:

20 registering an incarnation identifier of a most currently adopted spanning tree to the distributed dictionary.

27. The article of claim 24 wherein identifying the spanning tree as the new spanning tree comprises:

receiving the spanning tree at the particular node, said spanning tree including an incarnation identifier;

5 retrieving an incarnation identifier for an entry in the distributed dictionary corresponding to the spanning tree;

comparing the incarnation identifier of the spanning tree to the incarnation identifier for the entry; and

10 identifying the spanning tree as a new spanning tree if the incarnation identifier for the entry is older than the incarnation identifier for the spanning tree.

— 28. The article method of claim 23 wherein adopting the spanning tree comprises:

15 performing at least one iteration of an adoption task to disable any ports corresponding to links to be removed from a previous spanning tree; and

15 performing at least one iteration of the adoption task to enable any ports corresponding to links to be added from the previous spanning tree only after the links to be removed have been disabled.